REMARKS/ARGUMENTS

Claims 1-8, 11, 13-19, and 22-29 are pending in the present application. Claims 9, 12, 20, and 30 were canceled; and claims 1, 11, 13-15, 17-18, 22-23, and 27-28 were amended. Support for amendment of the claims can be found at least on pages 10 and 12 of the specification. Applicants are not conceding that the subject matter encompassed by those claims prior to cancellation and amendment is not patentable. Claims 9, 12, 20, and 30 were canceled and claims 1, 11, 13-15, 17-18, 22-23, and 27-28 were amended in this Amendment solely to facilitate expeditious prosecution of the remaining claims. Applicants respectfully reserve the right to pursue additional claims, including the subject matter encompassed by claims 9, 12, 20, and 30 as presented prior to this Amendment and the subject matter in the amended claims prior to amendment in one or more continuing applications. Reconsideration of the claims is respectfully requested.

I. Examiner Interview

Applicants thank Examiner Distefano for the courtesies extended to Applicants' representative during the June 12, 2008 telephone interview. During the interview, suggestions to amend the present application to overcome the 35 U.S.C. § 101 rejection were discussed. Examiner Distefano stated these amendments would overcome the 35 U.S.C. § 101 rejection. Suggestions to amend the present application to overcome the 35 U.S.C. § 103 rejection were also discussed. The substance of the interview is summarized in the remarks of the sections that follow.

II. 35 U.S.C. § 101

The Examiner has rejected claims 12-20 and 22-30 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

The Examiner states:

> Claims 12-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed to a data processing system" which may be interpreted to be purely software and the following "means" to be components of that software. Computer software fails to meet the 35 U.S.C. 101 requirement that the invention be a 'process, machine, manufacture, or composition of matter".

> Claims 22-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed to 'a computer program product in a computer readable medium". The "computer readable medium" may be interpreted to be an electrical signal, or other such intangible example, carrying thereon a computer program. Such forms of intangible media fail to meet the 35 U.S.C. 101 requirement that the invention be a "process, machine, manufacture, or composition of matter".

Office Action dated May 12, 2008, pages 2-3.

In response, claim 12 has been canceled, and claims 13-15, 17-18, and 20 have been amended to overcome the 35 U.S.C. § 101 rejection. Additionally, claims 22, 23, 27, and 28 have been amended to overcome the 35 U.S.C. § 101 rejection. Accordingly, these claims are now believed to be in condition for allowance.

III. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29 under 35 U.S.C. § 103 as being unpatentable over *Touboul*, U.S. Patent 6,125,390 (hereinafter *Touboul*), in view of *Agarwal*, U.S. Patent 6,305,010 (hereinafter *Agarwal*). This rejection is respectfully traversed.

In rejecting claim 1, the Examiner states:

> As per claims 1, 11, 12, and 22, Touboul teaches the following: monitoring operation of the program, (column 7, line 40), i.e. the agent 14 monitors the applications; determining whether an error has occurred based on the comparison, (column 7, lines 41-42), i.e. when an interrupt is generated, hooks or traps that interrupt and determines if there is an error condition. If so, the error is recorded as an alert which is reported to the monitor 2; responsive to an occurrence of the error, obtaining a solution for the error, (column 7, lines 44-46), i.e. the alert is sent by the agent to the monitor 2 and includes identification of the type of problem, the workstation on which it occurred, the name of the program which caused the error, and a recommended corrective action; and implementing the solution when the solution is obtained, (column 8, lines 58-64), i.e. one trigger is the function which is stored in a trigger library and can be called by the monitor 2 to be executed automatically as part of a procedure. The triggers can be called automatically in two cases, either in a correction procedure executed in response to an alert or in a scheduled procedure executed at a desired time as set up by the scheduling module 900. The examiner would like to further note Touboul's showing of Tables 2 and 3 in column 9 of their description which gives two possible lists of solutions for certain errors which may occur. However, Touboul does not

> explicitly teach a method where errors are detected by comparing an actual output with an expected output. Agarwal teaches the following: comparing an observed operation of the program with an expected operation of the program to form a comparison, (abstract), i.e. a type mismatch problem in computer programs is said to occur when there is a mismatch between the form or classification of a value encountered during program execution and that anticipated by the program.

Office Action dated May 12, 2008, pages 3-5.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are… determined; differences between the prior art and the claims at issue are… ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)). In this case, *Touboul* and *Agarwal* do not make the claimed invention obvious.

III.A. ***The Proposed Combination Does Not Teach All of the Features of Amended Claim 1***

Claim 1, as amended, is as follows:

> 1. A method in a data processing system for managing a program, the method comprising:
>
> monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier;

comparing an observed operation of the program with an expected operation of the program to form a comparison, wherein the expected operation is identified during compiling of the program;

determining whether an error has occurred based on the comparison;

responsive to an occurrence of the error, obtaining a solution for the error; and

implementing the solution when the solution is obtained.

The proposed combination, when considered as a whole, does not teach the claimed features of 1) monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier, and 2) comparing an observed operation of the program with an expected operation of the program to form a comparison, wherein the expected operation is identified during compiling of the program.

III.A.1. Monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier

The proposed combination fails to teach or suggest the monitoring step of amended claim 1. In rejecting claim 1, the Examiner cites to *Touboul* at column 7, line 40, which states:

> The agent **14** monitors the applications and the operating system and when an interrupt is generated, hooks or traps that interrupt and determines if there is an error condition. If so, the error is recorded as an alert which is reported to the monitor **2**. The alert is sent by the agent to the monitor **2** and includes identification of the type of problem, the workstation on which it occurred, the name of the program which caused the error, and a recommended corrective action. This recommended action can be modified by the administrator before it is executed.

The cited portion discloses monitoring applications and the operating system for an interrupt. *Touboul* discloses recording an alert when an interrupt is generated, and reporting the alert to a monitor. *Touboul* discloses the alert including recommended corrective action that can be modified by an administrator before the corrective action is executed. However, the cited

portion fails to disclose reading ahead of the execution of the *program* to identify behavior constructs and their identifiers. *Touboul* does not even address behavior constructs of a program and their identifiers. Thus, *Touboul* fails to teach or suggest "monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier."

The Examiner also cites to *Agarwal* at the abstract, which states:

> A type mismatch problem in computer programs is said to occur when there is a mismatch between the form or classification of a value encountered during program execution and that anticipated by the program. A method for repairing or testing for many type mismatch problems in programs works by transforming a binary representation of the program into a new binary in which the problem is fixed or identified. The fix or identification is implemented by converting code that operates on variables that can suffer a mismatch into code that correctly accounts for or tests for the mismatch. Static or dynamic correlation methods, and/or control and data flow graphs are used to track certain values, to determine where to install patches and how to adjust branch, jump and procedure call references after patch installation has shifted the target references.

The cited portion addresses testing for type mismatch problems between the form or classification of a value in programs. *Agarwal* teaches transforming a binary representation of the program into a new binary in which the problem is fixed or identified. Repairing the mismatch in *Agarwal* involves converting code that operates on variables that can suffer a mismatch into code that correctly accounts for or tests for the mismatch. *Agarwal* discloses a mismatch occurring *during* execution of a program. However, the cited portion fails to disclose monitoring operation of a program by reading *ahead* of the execution to identify behavior constructs and their identifiers. Furthermore, *Agarwal* fails to teach or suggest identification of behavior constructs and their identifiers in this or any other portion of the reference. Rather, *Agarwal* is concerned with a type mismatch of a value within a program. Thus, the cited art of *Touboul* and *Agarwal* fails to teach or suggest "monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier."

III.A.2. Comparing an observed operation of the program with an expected operation of the program to form a comparison, wherein the expected operation is identified during compiling of the program

The proposed combination fails to teach or suggest the comparing step of amended claim 1. In rejecting claim 1, the Examiner states that "*Touboul* does not explicitly teach a method where errors are detected by comparing an actual output with an expected output." Office Action dated May 12, 2008, page 4. The Examiner cites to *Agarwal* at the abstract, reproduced above.

As discussed above, *Agarwal* addresses testing for type mismatch problems of a value in programs. *Agarwal* teaches transforming a binary representation of the program into a new binary in which the problem is fixed or identified. Repairing and/or testing the mismatch in *Agarwal* involves converting code that operates on variables that can suffer a mismatch into code that correctly accounts for or tests for the mismatch. The process in *Agarwal* requires converting code that *can* suffer a mismatch into code that correctly *tests* for the mismatch. Code must be converted in order to even test for a type mismatch.

In contradistinction, the present invention claims a method for comparing an observed operation of the program with an expected operation of the program to form a comparison. The expected operation of the program is identified during *compiling* of the program. The present invention compares the observed operation of the program with the expected operation that was identified when the source code for the program was compiled. This occurs dynamically without the need for new or "fixed" code to be converted from the existing code and compared with the observed operation of the program. The present invention compares the observed operation with an expected operation identified previous to the execution of the program. *Agarwal* fails to teach or suggest comparing an observed operation with an expected operation identified during compiling of the program. Thus, the cited references fail to teach or suggest "comparing an observed operation of the program with an expected operation of the program to form a comparison, wherein the expected operation is identified during compiling of the program."

Therefore, the rejection of claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29 under 35 U.S.C. § 103 has been overcome.

III.B. ***Agarwal Teaches Away From The Presently Claimed Invention***

In addition, the Examiner has failed to establish a *prima facie* obviousness rejection against claim 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29 because no reason exists under the standards of *KSR Int'l.* to combine the references, considered as a whole. No reason exists to combine the references because *Agarwal* teaches away from claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29. A reference may be said to "teach away" from the claimed invention when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant. *In re Gurley*, 27 F.3d 551, 553, 31 U.S.P.Q.2D 1130, 1131 (Fed. Cir. 1995).

Agarwal teaches away from the presently claimed invention where *Agarwal* teaches converting code that operates on variables that can suffer a mismatch into code that correctly accounts for or tests for the mismatch. *Agarwal* leads away from the presently claimed invention because *Agarwal* requires converting code in order to test for a mismatch. The presently claimed invention compares the observed operation of the program with the expected operation identified during compiling of the program.

Because *Agarwal* teaches away from claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29, no motivation exists to combine *Touboul* and *Agarwal* as proposed by the Examiner. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection against claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29.

III.C. ***The Examiner Fails to Offer a Sufficient Reason to Modify the Reference***

In the case at hand, no *prima facie* obviousness rejection can be stated because the Examiner failed to state a sufficient reason to modify *Touboul* in view of *Agarwal* in light of the great differences between the cited art and amended claim 1. Specifically, as shown above, *Touboul* in view of *Agarwal* fails to teach or suggest the feature of 1) monitoring operation of the program, wherein monitoring the operation of the program comprises reading ahead of the execution to identify behavior constructs and their identifiers, wherein a profile of behavior constructs is created during compiling source code for the program, and wherein the profile of behavior constructs is stored in association with a unique identifier, and 2) comparing an observed operation of the program with an expected operation of the program to form a comparison, wherein the expected operation is identified during compiling of the program.

The Examiner failed to state a sufficient reason to modify *Touboul* in view of *Agarwal* because the Examiner's proposed reason for modifying the cited art provides no rational underpinning to support a legal conclusion of obviousness. Regarding a reason to modify *Touboul*, the Examiner states that:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the error detection method of Touboul with the comparison method of Agarwal. One of ordinary skill in the art would have been motivated to have made such a modification because both Touboul and Agarwal are analogous art in the field of error detection. Furthermore, Touboul teaches in column 7, lines 41-42, that "when an interrupt is generated, hooks or traps that interrupt and determines if there is an error condition. If so, the error is recorded as an alert which is reported to the monitor 2." Therefore, Touboul describes a desire to detect errors in an operating program where Agarwal teaches such a detection method. Thus, it would have been obvious to have utilized the detection through comparison method of Agarwal in the error determination method of Touboul.

Office Action dated May 12, 2008, page 5.

The Examiner offers an advantage as the stated reason for modifying the teachings of *Touboul* in view of *Agarwal* in the manner proposed by the Examiner. Specifically, the Examiner proposes modifying the cited art because *Touboul* describes a desire to detect errors in an operating program where *Agarwal* teaches such a detection method. However, the Examiner fails to provide a sufficient reason to modify *Touboul* in view of *Agarwal* because the Examiner merely offers a possible advantage for the modification without providing any reason for the modification. In particular, the Examiner does not provide any reason for modifying *Touboul* in view of *Agarwal* to detect errors where neither *Touboul* nor *Agarwal* teach or suggest all the features of amended claim 1. Thus, the Examiner's reason for modifying *Touboul* in view of *Agarwal* provides an insufficient basis for modifying the teachings of the cited art in the manner necessary to reach each and every feature of amended claim 1, especially in light of the large differences that exist between *Touboul* in view of *Agarwal* and amended claim 1.

For these reasons, the rejection of obviousness vis-à-vis amended claim 1 has been overcome. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection against claims 1-3, 6-8, 11-14, 17-19, 22-24, and 27-29.

III.D. Independent Claims

Independent claims 11 and 22 recite similar subject matter as that of claim 1. Therefore, at least by the reasons set forth above in regards to claim 1, claims 11 and 22 overcome the 35 U.S.C. § 103 rejection and are now in condition for allowance.

III.E. Dependent Claims

If an independent claim is non-obvious under 35 U.S.C. §103, then any claim depending therefore is also non-obvious by virtue of their dependency. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Claims 2-8, 13-19, and 23-29 depend from claims 1, 11, and 22. Applicants have already demonstrated claims 1, 11, and 22 are not obvious and are therefore in condition for allowance. Therefore, at least by virtue of their dependence on claims 1, 11, and 22, claims 2-8, 13-19, and 23-29 are not obvious over *Touboul* in view of *Agarwal*.

As shown above, *Touboul* in view of *Agarwal* fails to teach or suggest all of the features of claims 1-8, 11, 13-19, and 22-29. Additionally, *Agarwal* teaches away from the presently claimed invention. Therefore, the proposed combination and modification of the cited references when considered together as a whole does not teach or suggest all of the features of claims 1-8, 11, 13-19, and 22-29. Therefore, the Examiner has failed to state a *prima facie* obviousness rejection against these claims.

IV. Conclusion

It is respectfully urged that the subject application is patentable over *Touboul* in view of *Agarwal* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: June 16, 2008

Respectfully submitted,

/Mari A. Stewart/

Mari A. Stewart
Reg. No. 50,359
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

MAS/sbf